Three-dimensional finite difference saturated-unsaturated flow modeling with nonorthogonal grids using a coordinate transformation method

AUTHOR(S):

An, Hyunuk; Ichikawa, Yutaka; Tachikawa, Yasuto; Shiiba, Michiharu

CITATION:

An, Hyunuk ...[et al]. Three-dimensional finite difference saturated-unsaturated flow modeling with nonorthogonal grids using a coordinate transformation method. Water Resources Research 2010, 46: W11521.

ISSUE DATE:

2010-11

URL:

http://hdl.handle.net/2433/134610

WATER RESOURCES RESEARCH, VOL. ???, XXXX, DOI:10.1029/,

Three-dimensional finite-difference saturated–unsaturated flow modeling with non-orthogonal grids, using a coordinate transformation method

Hyunuk An[1], Ichikawa Yutaka[2], Tachikawa Yasuto[3], and Shiiba Michiharu[3]

Hyunuk An, Graduate School of Engineering, Kyoto University, Nishikyoku, Kyoto, 615-8540, Japan. (an@hywr.kuciv.kyoto-u.ac.jp)

Yutaka Ichikawa, Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Kofu, Yamanashi, 400-8511, Japan. (yichikawa@yamanashi.ac.jp)

Yasuto Tachikawa, Faculty of engineering, Kyoto University, Nishikyoku, Kyoto, 615-8540, Japan. (tachikawa@mbox.kudpc.kyoto-u.ac.jp)

Michiharu Shiiba, Faculty of engineering, Kyoto University, Nishikyoku, Kyoto, 615-8540, Japan. (shiiba@mbox.kudpc.kyoto-u.ac.jp)

[1]Graduate School of Engineering, Kyoto

Abstract. Study of the saturated-unsaturated flow in porous media is of interest in many branches of science and engineering. Among the various numerical simulation methods available, the finite-difference method is advantageous because it offers simplicity of discretization. This method has been widely used for simulating saturated-unsaturated flows. However, the simulation of geometrically complex flow domains requires the use of high-resolution grids in conventional finite-difference models because conventional finite-difference discretization assumes an orthogonal coordinate system. This makes a finite-difference model computationally less efficient than other numerical models that can treat non-orthogonal grids, such as the finite-element model and finite-volume model. To overcome this disadvantage, we use a coordinate transformation method and develop a multidimensional finite-difference model for simulating saturated-unsaturated flows; this model can treat non-orthogonal grids. The cross-derivative terms derived by the coordinate transformation method are evaluated explicitly for ease of coding. Therefore, a 7-point stencil is used for implicit terms in the iterative calculation, as in the case of con-

University, Kyoto, Japan.

[2]Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Yamanashi, Japan.

[3]Faculty of engineering, Kyoto University, Kyoto, Japan.

ventional finite-difference models with an orthogonal grid. We assess the performance of the proposed model by carrying out test simulations. We then compare the simulation results with dense-grid solutions in order to evaluate the numerical accuracy of the proposed model. To examine the performance of the proposed model, we draw a comparison between the simulation results obtained using the proposed model and the results obtained by using (1) a model in which all terms are considered fully implicitly, (2) a finite-element model, and (3) a conventional finite-difference model with a high-resolution orthogonal grid.

1. Introduction

Saturated-unsaturated flow through porous media is an important research topic in water resources engineering, agricultural engineering, contaminant tracing, rainfall-runoff modeling, etc. Several analytical solutions have proposed for the governing equation of saturated-unsaturated flows through porous media [*Philip*, 1957; *Parlange*, 1972; *Broadbridge & White*, 1988; *Hogarth & Parlange*, 2000; *Menziani et al.*, 2007]; however, these solutions are generally obtained under simple initial and boundary conditions. Hence, numerical models are usually used to investigate saturated-unsaturated flows in porous media, where analytical solutions are not appropriate. Over the past three decades, many numerical models, including finite-difference models and finite-element models, have been developed for simulating saturated-unsaturated flows [*Celia et al.*, 1990; *Clement et al.*, 1994; *Forsyth*, 1995; *Tocci et al.*, 1997; *Šimůnek et al.*, 1999; *Jones & Woodward*, 2001]. Other approaches such as the finite-volume approach, mixed finite-element approach, and Eulerian-Lagrangian approach have also been proposed for simulating saturated-unsaturated flows [*Manzini & Ferraris*, 2004; *Bause & Knabner*, 2004; *Farthing et al.*, 2003; *Huang et al.*, 1994].

The finite-difference model has distinct advantages over other models because of its simplicity of discretization; further, this mode is easy to interpret and code. The finite-difference model has been widely used for simulating saturated-unsaturated flow [*Freeze*, 1971; *Cooley*, 1971; *Dane & Mathis*, 1981; *Haverkamp & Vauclin*, 1981; *Clement et al.*, 1994; *Weeks et al.*, 2004; *Dogan & Motz*, 2005]. However, finite-difference do not give an accurate representation of geometrically complex flow domains with low resolution, espe-

cially in multidimensional simulations. High-resolution orthogonal grids are required for domains with inherent curvilinear features, such as foundation fits [*Jie et al.*, 2004], embarkment dams [*Billstein et al.*, 1999], and shallow groundwater flows with a curvilinear boundary [*Chapman & Ong*, 2006; *Liang et al.*, 2009]. Finite-difference models are computationally less efficient for the simulation of such domains than are finite-element and finite-volume models, which can be used to treat non-orthogonal grids. The finite-element and finite-volume models can accommodate curvilinear domains with comparatively low resolution grids because they can be used to discretize the governing equation of saturated-unsaturated flows without the need for using an orthogonal coordinate system [*Šimůnek et al.*, 1999; *Manzini & Ferraris*, 2004]. If the domain shape is extremely complex, this inefficiency would be significantly increased. Furthermore, the principal axes of anisotropy are typically aligned with the orthogonal axis in finite-difference models, which rely on orthogonal grids. In the case of these models, it is mandatory for the principal axes of anisotropy to be oriented in the same direction throughout the flow domain.

To overcome the disadvantages associated with finite-difference modeling, we applied a coordinate transformation method to a saturated-unsaturated flow equation. This method is based on tensor analysis and has been commonly applied to the general Navier-Stokes equation in computational fluid dynamics (CFD) [*Maliska & Raithby*, 1984; *Hsu et al.*, 1998; *Wesseling*, 2001; *Chung*, 2002]; it has also been used for describing the circulation and transport in estuaries and oceans [*Murray & Reason*, 2001; *Chen*, 2004]. Further, this method has been successfully applied to heat transport [*Lakner & Plazl*, 2008; *Ruhaak et al.*, 2008] and groundwater modeling [*Kinouchi et al.*, 1991; *Koo & Leap*, 1998a, b; *Jie et al.*, 2004; *Ruhaak et al.*, 2008]. *Koo & Leap* [1998a, b] proposed a finite-difference-based

model for groundwater flow, in which they used a successive over-relaxation (SOR) method to solve the system of equations. The cross-derivative terms were evaluated at the previous iteration level for convergence stability; this is because the matrix of the system equation should be diagonally dominant when the SOR method is used. *Jie et al.* [2004] used a coordinate-transform finite-difference model and carried out steady-state groundwater modeling to analyze the seepage flow in the case of a foundation fit, a lock foundation, and an embarkment dam. *Ruhaak et al.* [2008] applied the coordinate transformation method to a finite-volume model for simulating heat transport and groundwater flow. They solved the cross-derivative terms explicitly and the other terms implicitly. The abovementioned three models can only be used for simulating saturated flows. *Kinouchi et al.* [1991] applied the coordinate transformation method to a finite-difference model for simulating a two-dimensional unsaturated flow in porous media. They solved the pressure-head-based Richards' equation, in which the transformed equation takes a nonconservative form because the mesh skewness tensor is cast outside the differential operator. Apart from the abovementioned example, there are very few reports on the use of the coordinate transformation method for simulating saturated-unsaturated flows. However, we believe that by using this method, the constraints on the representation of a curvilinear shape can be relaxed, and the principal axes of anisotropy can be aligned with the curvilinear surface, because of which the orientation of the axes changes gradually and continuously throughout the flow domain in a saturated-unsaturated flow model.

Because saturated-unsaturated flow equation are highly nonlinear, implicit temporal discretization and iterative procedures are required to ensure numerical stability. Coordinate transformation represents diffusion with cross-derivative terms. Therefore, the

transformed equation requires a 19-point stencil instead of a 7-point stencil, which is used in the case of a conventional finite-difference model with an orthogonal grid. The complexity of the algorithm and the additional storage makes this scheme less attractive. Furthermore, implicit evaluation of all the terms can be inefficient in the case of quasi-orthogonal grids because the cross-derivative terms make minor contributions to the overall system. Therefore, we implement an approach in which the cross-derivative terms are evaluated at the previous iteration level, while the other terms are evaluated at the current iteration level. In this approach, the 7-point stencil is calculated implicitly by iterative calculations, as in the case of the conventional finite-difference model. In this manner, the proposed scheme can be used to treat curvilinear coordinate systems without compromising on the computational efficiency and simplicity of discretization.

The remainder of this paper is organized as follows. In Section 2, the governing equation for a saturated-unsaturated flow in porous media is presented. This physical-space equation is then transformed into a computational-space equation. In Section 3, spatial and temporal discretization and a method for evaluating cross-derivative terms and metrics are described. In Section 4, the details of four test simulations carried out to verify the performance and accuracy of the proposed model are described. In the first test simulation, a steady-state problem is considered for studying the applicability of the proposed model with a highly skewed grid. In the second test simulation, an infiltration problem is considered for evaluating the effect of mesh skewness on the performance and accuracy of the proposed model. The third test is the simulation of a transient, variably saturated flow and is carried out to demonstrate the performance of the proposed model in the case of a non-rectangular flow domain. The fourth test is the simulation of

a three-dimensional rainfall runoff and is carried out to show the performance of the proposed model for a three-dimensional curvilinear-shaped flow domain. The results of these simulations are compared with a numerical solution of the conventional finite-difference model that utilizes a high-resolution, stepwise orthogonal grid and a finite-element model HYDRUS [*Šimůnek et al.*, 1999].

2. Governing equation

Richards' equation, which is typically used to simulate saturated-unsaturated flows, is written as

$$\frac{\partial \theta(\psi)}{\partial t} = \nabla \cdot K(\psi) \nabla \psi + \frac{\partial K(\psi)}{\partial z}, \tag{1}$$

where ψ is the pressure head; θ, the volumetric moisture content; K, the hydraulic conductivity; t, the time; and z, the vertical dimension, which is assumed to be positive in the upward direction. Further, appropriate constitutive relationships are assumed to exist between θ and ψ and between ψ and K . The source/sink term has been ignored for simplicity. Equation (1) includes both θ and ψ and is thus called the mixed form of Richards' equation. This form is generally considered to be advantageous over the other two forms, namely, the ψ-based and θ-based forms, because of its perfect mass balance [*Celia et al.*, 1990; *Mansell et al.*, 2002].

Coordinate transformation is commonly used in CFD for the general Navier?Stokes equation. This technique can be used to transform a curvilinear grid into a rectangular grid, as shown in Fig. 1. A curvilinear coordinate system in physical space, (x_1, x_2, x_3) = (x, y, z), can be transformed into a new coordinate system in computational space, (ξ_1, ξ_2, ξ_3)=(ξ, η, ζ).

The summation form of Eq. (1) is

$$\frac{\partial\theta}{\partial t} = \sum_{r=1}^{3}\sum_{s=1}^{3}\frac{\partial}{\partial x_r}\left\{K_{r,s}\frac{\partial\psi}{\partial x_s}\right\} + \sum_{r=1}^{3}\frac{\partial K_r}{\partial x_3}, \tag{2}$$

where $K_{r,s}$ is an element of the hydraulic conductivity tensor. This equation is transformed into

$$\frac{1}{J}\frac{\partial\theta}{\partial t} = \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K\frac{\partial\psi}{\partial\xi_q}\right\} + \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK\right), \tag{3}$$

where

$$J = \frac{\partial(\xi_1,\xi_2,\xi_3)}{\partial(x_1,x_2,x_3)}, \tag{4}$$

$$G^{p,q}K = \sum_{r=1}^{3}\sum_{s=1}^{3}\frac{K_{r,s}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\xi_q}{\partial x_s}, \qquad H^pK = \sum_{r=1}^{3}\frac{K_{r,3}}{J}\frac{\partial\xi_p}{\partial x_3}. \tag{5}$$

J is the Jacobian determinant that represents the ratio of the control volume in physical space to that in computational space. $G^{p,q}$ and H^p are operators that represent the mesh skewness tensor [*Zang et al.*, 1994]. Detailed evaluation of J and the metrics, $\partial\xi_p/\partial x_r$, will be shown in Section 3.2. The derivation of Eq. (3) is given in detail in the Appendix. Eq. (3) is mass conservative because the mesh skewness tensors lie inside the differential operators [*Morinishi et al.*, 1998; *Kogaki*, 1999; *Koo & Leap*, 1998a].

3. Numerical modeling

3.1. Temporal discretization

The backward Euler scheme used in this study is one of the most widely used time approximation schemes for Richards' equation. The equation system is nonlinear owing to the nonlinear dependence of θ on ψ, and hence, iterative calculations and linearization are necessary. Although several iterative methods (e.g., Picard and Newton iteration methods, fast secant methods, and relaxation methods) have been proposed [*Bergamaschi & Putti*,

1999; *Fassino & Manzini*, 1999; *Kavetski et al.*, 2002], we use the Picard method because it is simple and delivers a satisfactory performance [*Paniconi et al.*, 1991; *Paniconi & Putti*, 1994; *Lehmann & Ackerer*, 1998]. The backward Euler approximation and Picard linearization of the three-dimensional Eq. (3) can be expressed as

$$\frac{1}{J}\frac{\theta^{n+1,m+1}-\theta^{n}}{\Delta t} = \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\} + \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1,m}\right), \tag{6}$$

where the superscripts n and m denote the time level and iteration level, respectively. The terms with $p = q$ in Eq. (6) represent the contributions made by the normal derivatives, and the other terms ($p \neq q$) represent the contributions made by the cross-derivatives.

The moisture content at the new time step and iteration level ($\theta^{n+1,m+1}$) are replaced with the Taylor series expansion with respect to ψ, about the expansion point $\psi^{n+1,m}$, as follows:

$$\theta^{n+1,m+1} = \theta^{n+1,m} + \left.\frac{d\theta}{d\psi}\right|^{n+1,m}(\psi^{n+1,m+1}-\psi^{n+1,m}) + O(\delta^2). \tag{7}$$

By neglecting the higher-order terms in Eq. (7) and substituting this equation into Eq. (6), we get

$$\frac{C^{n+1,m}}{J}\frac{\psi^{n+1,m+1}-\psi^{n+1,m}}{\Delta t} + \frac{1}{J}\frac{\theta^{n+1,m}-\theta^{n}}{\Delta t} = \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\} + \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^{p}K^{n+1,m}\right), \tag{8}$$

where $C(= d\theta/d\psi)$ is the specific moisture capacity function.

3.2. Finite-difference discretization

Because the transformed equation gives cross-derivative terms, as in Eq. (6), a 19-point stencil is required to solve the equation fully implicitly. It is not feasible to consider all the

terms implicitly because the cross-derivative terms usually make minor contributions to the overall system. Moreover, it is difficult to code a simultaneous equation with a 19-point stencil. Hence, we propose to evaluate the cross-derivative terms at the previous iteration level. The cross-derivative terms are solved implicitly in the temporal discretization and explicitly in the spatial discretization.

In the computational space, the grid is orthogonal and all grid sizes ($= \Delta\xi_p$) are set to 1. The grid sizes have no virtual influence on the simulation results. Hence, computational grids of any size can be used. However, when a grid with a size other than 1 is used, a round-off error is introduced in the results and dividing by 1 is very convenient. Hence, the grid size is usually set to 1. The finite-difference approximation of Eq. (8) can be given as

$$\frac{C^{n+1,m}_{i,j,k}}{J}\frac{\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m}_{i,j,k}}{\Delta t}+\frac{1}{J}\frac{\theta^{n+1,m}_{i,j,k}-\theta^{n}_{i,j,k}}{\Delta t} \tag{9}$$
$$=\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\}_{i,j,k}+\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right)_{i,j,k},$$

where the first term on the right-hand side of Eq. (9) is evaluated partially implicitly by

$$\sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K^{n+1,m}\left.\frac{\partial\psi}{\partial\xi_q}\right|^{n+1,m+1}\right\}_{i,j,k} \tag{10}$$
$$\begin{aligned}
=\ & \left\{G^{1,1}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m+1}_{i+1,j,k}-\psi^{n+1,m+1}_{i,j,k})\right.\\
& \left.-G^{1,1}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i-1,j,k})\right\}\\
+\ & \left\{G^{2,2}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m+1}_{i,j+1,k}-\psi^{n+1,m+1}_{i,j,k})\right.\\
& \left.-G^{2,2}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i,j-1,k})\right\}\\
+\ & \left\{G^{3,3}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m+1}_{i,j,k+1}-\psi^{n+1,m+1}_{i,j,k})\right.\\
& \left.-G^{3,3}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m+1}_{i,j,k}-\psi^{n+1,m+1}_{i,j,k-1})\right\}
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{2}\left\{G^{1,2}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j+1,k} - \psi^{n+1,m}_{i+1/2,j-1,k})\right.\\
&\left. - G^{1,2}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j+1,k} - \psi^{n+1,m}_{i-1/2,j-1,k})\right\}\\
&+ \frac{1}{2}\left\{G^{1,3}K^{n+1,m}_{i+1/2,j,k}(\psi^{n+1,m}_{i+1/2,j,k+1} - \psi^{n+1,m}_{i+1/2,j,k-1})\right.\\
&\left. - G^{1,3}K^{n+1,m}_{i-1/2,j,k}(\psi^{n+1,m}_{i-1/2,j,k+1} - \psi^{n+1,m}_{i-1/2,j,k-1})\right\}\\
&+ \frac{1}{2}\left\{G^{2,1}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i+1,j+1/2,k} - \psi^{n+1,m}_{i-1,j+1/2,k})\right.\\
&\left. - G^{2,1}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i+1,j-1/2,k} - \psi^{n+1,m}_{i-1,j-1/2,k})\right\}\\
&+ \frac{1}{2}\left\{G^{2,3}K^{n+1,m}_{i,j+1/2,k}(\psi^{n+1,m}_{i,j+1/2,k+1} - \psi^{n+1,m}_{i,j+1/2,k-1})\right.\\
&\left. - G^{2,3}K^{n+1,m}_{i,j-1/2,k}(\psi^{n+1,m}_{i,j-1/2,k+1} - \psi^{n+1,m}_{i,j-1/2,k-1})\right\}\\
&+ \frac{1}{2}\left\{G^{3,1}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i+1,j,k+1/2} - \psi^{n+1,m}_{i-1,j,k+1/2})\right.\\
&\left. -G^{3,1}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i+1,j,k-1/2} - \psi^{n+1,m}_{i-1,j,k-1/2})\right\}\\
&+ \frac{1}{2}\left\{G^{3,2}K^{n+1,m}_{i,j,k+1/2}(\psi^{n+1,m}_{i,j+1,k+1/2} - \psi^{n+1,m}_{i,j-1,k+1/2})\right.\\
&\left. -G^{3,2}K^{n+1,m}_{i,j,k-1/2}(\psi^{n+1,m}_{i,j+1,k-1/2} - \psi^{n+1,m}_{i,j-1,k-1/2})\right\},
\end{aligned}
$$

and the second term on the right-hand side of Eq. (9) is given as

$$
\begin{aligned}
&\sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK^{n+1,m}\right)_{i,j,k} \qquad (11)\\
&= \left(H^1K^{n+1,m}\right)_{i+1/2,j,k} - \left(H^1K^{n+1,m}\right)_{i-1/2,j,k}\\
&+ \left(H^2K^{n+1,m}\right)_{i,j+1/2,k} - \left(H^2K^{n+1,m}\right)_{i,j-1/2,k}\\
&+ \left(H^3K^{n+1,m}\right)_{i,j,k+1/2} - \left(H^3K^{n+1,m}\right)_{i,j,k-1/2},
\end{aligned}
$$

where the subscripts i, j, and k denote the spatial coordinates along the ξ_1, ξ_2, and ξ_3 axes, respectively, in the computational space. The 1/2 coefficients are derived from the terms of $\partial\psi/\partial\xi_q$ on $G^{p,q}$, where p is not equal to q. For example, where $p = 1$ and $q = 2$

in the fourth term on the right-hand side of Eq. (10), $(\partial\psi/\partial\xi_q)_{i+1/2,j,k}$ is evaluated as $(\psi^{n+1,m}_{i+1/2,j+1,k}-\psi^{n+1,m}_{i+1/2,j-1,k})/2\Delta\xi_2$. Because the grid size in the computational space is equal to 1, the term yields $(\psi^{n+1,m}_{i+1/2,j+1,k}-\psi^{n+1,m}_{i+1/2,j-1,k})/2$. The first, second, and third terms on the right-hand side of Eq. (10), including ψ^{m+1}, are calculated implicitly, and the other terms on the right-hand side of Eq. (10), including ψ^{m}, are calculated explicitly.

If all the terms are calculated implicitly, the 19 unknown variables must be included in the linearized simultaneous equation, as shown in Fig. 2-(a). In contrast, in the proposed scheme, the linearized simultaneous system includes only seven unknown variables ($\psi^{n+1,m+1}_{i+1,j,k}$, $\psi^{n+1,m+1}_{i,j+1,k}$, $\psi^{n+1,m+1}_{i,j,k+1}$, $\psi^{n+1,m+1}_{i,j,k}$, $\psi^{n+1,m+1}_{i,j,k-1}$, $\psi^{n+1,m+1}_{i,j-1,k}$, and $\psi^{n+1,m+1}_{i-1,j,k}$) from Eq. (10), as shown in Fig. 2-(b). This makes the proposed scheme simpler and more efficient when cross-derivative terms are not dominant in the system. The cross-derivative terms make negligible contributions when the grid is not highly skewed. However, if the grid is highly skewed, the cross-derivative terms make a large contribution to the overall system, and evaluating the cross-derivative terms might require a small time-step duration. This issue will be discussed in Section 4.2. Irrespective of the scheme used, the 7-point stencil $G^{p,q}K$ is utilized, as shown in Fig. 2-(c).

The hydraulic conductivity and ψ of the boundary between the adjacent nodes used the arithmetic mean are given as

$$(K_{r,s})_{i\pm1/2,j,k} = \frac{1}{2}\left\{(K_{r,s})_{i\pm1,j,k}+(K_{r,s})_{i,j,k}\right\}, \tag{12}$$

$$\psi_{i\pm1/2,j,k} = \frac{1}{2}\left(\psi_{i\pm1,j,k}+\psi_{i,j,k}\right). \tag{13}$$

The linearized simultaneous system can be solved by using matrix solvers such as those based on LU decomposition or the preconditioned conjugate gradient method. In this study, we use a library of iterative solvers for linear systems (LIS) developed by *Kotake-*

mori et al. [2005]. The LIS provides several preconditioners and iterative solvers for linear systems. On the basis of the results of the test simulations, we select a pair of symmetric successive over-relaxation (SSOR) preconditioners and a biconjugate gradient method; this pair is faster than the other pairs provided by the LIS.

The iteration of Eq. (9) is continued until the difference between the calculated values of ψ or θ of two successive iteration levels becomes less than the user-specified tolerance, i.e., until the following inequality is satisfied for all cells:

$$|\psi^{n+1,m+1} - \psi^{n+1,m}| \leq \delta_\psi, \tag{14}$$

$$|\theta^{n+1,m+1} - \theta^{n+1,m}| \leq \delta_\theta, \tag{15}$$

where δ_ψ and δ_θ are the convergence tolerances. The results of a previous study show that the θ-based convergence tolerance of Eq. (13) can help in converging the model when simulating infiltration problems using dried soil [*Huang et al.*, 1996]. $\delta_\psi = 0.001$ m and $\delta_\theta = 0.0001$ are used in all the test simulations performed in this study except in Test 1, where steady-state conditions are simulated. The convergence tolerance in Test 1 is $\delta_\psi = 10^{-4}$ m.

3.3. Metrics evaluation

The metrics set and Jacobian determinant are defined as

$$\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta), \quad \xi_y = J(z_\eta x_\zeta - z_\zeta x_\eta), \quad \xi_z = J(x_\eta y_\zeta - x_\zeta y_\eta), \tag{16}$$

$$\eta_x = J(y_\zeta z_\xi - y_\xi z_\zeta), \quad \eta_y = J(z_\zeta x_\xi - z_\xi x_\zeta), \quad \eta_z = J(x_\zeta y_\xi - x_\xi y_\zeta),$$

$$\zeta_x = J(y_\xi z_\eta - y_\eta z_\xi), \quad \zeta_y = J(z_\xi x_\eta - z_\eta x_\xi), \quad \zeta_z = J(x_\xi y_\eta - x_\eta y_\xi),$$

$$\frac{1}{J} = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi),$$

where $(\xi_p)_{x_r} = \partial\xi_p/\partial x_r$ and $(x_r)_{\xi_p} = \partial x_r/\partial\xi_p$. The Jacobian determinant J is used at a node, as shown in Fig. 2-(a), and the mesh skewness tensor G is used between adjacent nodes, as shown in Fig. 2-(c). Because the size of the computational cell is set to 1, $\partial x_r/\partial\xi_p$ at a node $(i,\, j,\, k)$ is evaluated as

$$\left.\frac{\partial x_r}{\partial \xi_1}\right|_{i,j,k} = \frac{(x_r)_{i+1,j,k} - (x_r)_{i-1,j,k}}{2}, \qquad \left.\frac{\partial x_r}{\partial \xi_2}\right|_{i,j,k} = \frac{(x_r)_{i,j+1,k} - (x_r)_{i,j-1,k}}{2}, \qquad (17)$$
$$\left.\frac{\partial x_r}{\partial \xi_3}\right|_{i,j,k} = \frac{(x_r)_{i,j,k+1} - (x_r)_{i,j,k-1}}{2}.$$

For evaluating the metrics at a boundary node, we assume that an additional row of ghost nodes with zero thickness is added to the outside nodes, as shown in Fig. 3 [*Peric*, 1994]. These ghost nodes are used only for evaluating the metrics at the boundary nodes. For example, if $i = 0$, which is at the real boundary node, $\partial x_r/\partial\xi_1$ of Eq. (17) cannot be calculated because $(x_r)_{-1,j,k}$ is not defined. However, when using the ghost nodes $(x_r)_{-1,j,k} = (x_r)_{0,j,k}$, the value of $\partial x_r/\partial\xi_1$ can be easily evaluated. The other metrics at the boundary nodes, such as $\partial x_r/\partial\xi_2$ and $\partial x_r/\partial\xi_3$, are evaluated in a similar manner.

The metrics between adjacent nodes for calculating G are evaluated as

$$\left.\frac{\partial x_r}{\partial \xi_1}\right|_{i+1/2,j,k} = (x_r)_{i+1,j,k} - (x_r)_{i,j,k}, \qquad \left.\frac{\partial x_r}{\partial \xi_2}\right|_{i,j+1/2,k} = (x_r)_{i,j+1,k} - (x_r)_{i,j,k}, \qquad (18)$$
$$\left.\frac{\partial x_3}{\partial \xi_r}\right|_{i,j,k+1/2} = (x_r)_{i,j,k+1} - (x_r)_{i,j,k}.$$

3.4. Boundary condition

There are two main types of boundary conditions, namely, the Dirichlet boundary condition and the Neumann boundary condition. Dirichlet boundaries are treated in the same manner as they are in the case of the other finite-difference models (predetermined ψ). Under the Neumann boundary condition, the flux must be transformed. If the flux $q_{p'}$ is directed from outside to the (i, j, k)-node and the element of $q_{p'}$ is (q_1, q_2, q_3) in the

physical space, $q_{p'}$ can be transformed into

$$q_{p'} = \sum_{r=1}^{3} \frac{q_r}{J} \frac{\partial \xi_{p'}}{\partial x_r}, \tag{19}$$

where p' is 1, 2, and 3 when the flux passes through the surfaces of $\xi_2\xi_3$, $\xi_3\xi_1$, and $\xi_1\xi_2$, respectively, as shown in Fig. 4-(a), (b), and (c).

The seepage face is treated as follows [*Clement et al.*, 1994]. If the location of the seepage face is known, all the nodes along the seepage face can be treated as Dirichlet boundaries ($\psi = 0$). However, the exact range of the seepage face is usually unknown until the problem is solved. Hence, the location of the seepage face must be determined iteratively. In the first iteration, the location of the seepage face is approximated to the same location as that in the previous time step. If the guess the location of the seepage face is correct, the flux of the nodes along the seepage face is assumed to be outward, and the values of ψ at the boundary nodes above the seepage face are negative. If the nodes where $\psi = 0$ have a net inward flux, the nodes where the flux is inward are assumed to be no-seepage faces. If the nodes above the seepage face have positive values of ψ, it implies that these nodes should be set as seepage faces. On the basis of these principles, the location of the seepage face is determined iteratively. Details of the determination procedure are provided in *Neuman* [1973].

3.5. Grid generation

Grid generation is an important field of research. Most previous studies on the generation of a block-structured grid involved the use of algebraic methods or partial differential equation (PDE) methods. Although algebraic methods have a major advantage in that they facilitate rapid computation, the obtained grids have low smoothness; hence, PDE

methods are more preferred over algebraic methods [*Koo & Leap*, 1998a]. The most widely used PDE method is that based on the Poisson equation with specified control functions. These specified control functions allow the inner grid nodes to be concentrated in specific regions or orthogonally positioned at the boundaries. *Thompson et al.* [1999] presented a comprehensive review of various numerical grid generation methods and a detailed procedure for the generation of block-structured grids. In this study, an algebraic method is used for generating grids for Test 1∼3, and a Poisson equation system is used for Test 4.

4. Verification

Test simulations were carried out to verify the performance of the proposed model. The first and second test simulations were carried out to evaluate the model performance for different mesh skewness. The third and fourth test simulations were carried out to evaluate the model performance for non-rectangular and curvilinear flow domains. The results of the first, second, and third test simulations were compared with an exact solution or a numerical solution with higher resolution. For the sake of comparison, test simulations were also carried out using the conventional finite-difference model HYDRUS, and the resulting performance of HYDRUS was compared with that of the proposed model.

The mass balance error and relative error are estimated in the test simulations. The mass balance error is given by [*Celia et al.*, 1990]

$$\text{Mass Balance Error} = \left| 1 - \frac{\text{Total additional mass in the domain}}{\text{Total net flux into the domain}} \right|, \tag{20}$$

where the total additional mass in the domain is the difference between the mass measured at any instant t and the initial mass in the domain; the total net flux into the domain is the flux balance integrated with respect to time up to time t. The relative error is

estimated in the manner described in *Manzini & Ferraris* [2004]:

$$\text{Relative Error} = \frac{\sqrt{\sum_i (\bar{\psi}_i - \psi_i)^2 / J_i}}{\sqrt{\sum_i \bar{\psi}_i{}^2 / J_i}}, \tag{21}$$

where ψ_i is the ith node solution obtained using the models, and $\bar{\psi}_i$ is the ith node exact solution for the pressure head. $1/J_i$ is the volume represented by the ith node.

Van Genuchten [1980]'s equation for the soil water retention curve and *Mualem* [1976]'s equation for the unsaturated hydraulic conductivity function are used in this study. The soil water retention curve is given by

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} = \left\{ \frac{1}{1 + (\alpha|\psi|)^n} \right\}^{1-1/n}, \tag{22}$$

where S_e is the effective saturation; θ_r and θ_s are the residual and saturated water contents, respectively. α and n are van Genuchten parameters whose values depend on the soil properties. The nonlinearity of Richards' equation is attributed to the nonlinear dependence of θ on ψ, which is determined by α and n, in the case of the van Genuchten model. In general, it should be noted that a large α and n indicate high nonlinearity. Following Mualem's model, the unsaturated hydraulic conductivity function is given by

$$K = K_s S_e^{1/2} \left\{ 1 - (1 - S_e^{n/(n-1)})^{1-1/n} \right\}^2, \tag{23}$$

where K_s is the saturated hydraulic conductivity. In all the test simulations, we consider homogeneous and isotropic soil domains and use the soil properties listed in Table 1, by referring to *Carsel & Parrish* [1988]. These values represent the average of the selected soil water retentions and hydraulic conductivities for major soil textural groups.

The time-step durations are adjusted automatically on the basis of the number of iterations required for convergence at the previous time step [*Paniconi & Putti*, 1994]. The time-step duration cannot be less than a preselected minimum value and cannot exceed a

maximum duration. If the number of iterations required for convergence is less than N_m, the time-step duration for the next time step is multiplied by C_m, a predetermined value greater than 1. If the number of iterations is greater than N_r, the time-step duration for the next time step is multiplied by C_r, a preselected value less than 1. If the number of iterations becomes greater than a prescribed N_b value, the iterative process for the time level is terminated. Subsequently, the time-step duration is multiplied by C_b, a predetermined value that is less than 1, and the iterative process is restarted. These time-step duration control factors normally need to be adjusted on the basis of the difficulty and size of a problem, so that the iterative scheme delivers a satisfactory performance. In the test simulations presented in this paper, $C_m = 1.3$, $C_r = 0.7$, $C_b = 0.33$, $N_m = 4$, $N_r = 7$ and $N_b = 20$ are used.

4.1. Test 1: steady-state simulation with a highly skewed mesh

To test the model performance, a two-dimensional steady-state simulation with a highly skewed mesh was carried out using *Kershaw*'s [1981] mesh, which is often used for testing the accuracy of a diffusion equation. A square domain (side: 1 m) based on two types of meshes was considered, as shown in Fig. 5. The 20 × 20 mesh was more severely skewed than the 10 × 10 mesh. The silt soil properties listed in Table 1 were considered. The top and bottom boundaries were $\psi_{top} = -0.5$ m and $\psi_{bot} = 0.5$ m, respectively, and both the side boundaries were assumed to be no-flow boundaries. In fact, the correct solution was a linear distribution $\psi(z) = 0.5 - z$ m.

Figs. 6-(a) and (b) show the isolines of the pressure head calculated by using the proposed model. The isoline plots of the pressure head are straight lines that are independent of the mesh shape. Even with the highly skewed mesh, the model gave an accurate repre-

sentation of the pressure head distribution. The root-mean-square errors in the pressure head are shown in Figs. 6-(c) and (d). The overall error obtained with the 20 × 20 mesh was larger than that obtained with the 10 × 10 mesh, even though the former mesh had a finer spatial resolution. In general, the accuracy of a model using a non-orthogonal grid is influenced by the mesh quality: when the non-orthogonality is high, the numerical accuracy is reduced because the truncation error becomes large [*Mastin*, 1998]. Therefore, to achieve high numerical accuracy, it is preferable to avoid highly skewed grids.

4.2. Test 2: unsteady-state simulation to investigate the non-orthogonality effect

This test simulation is carried out to investigate the effect of grid skewness on the performance and accuracy of the proposed model. As discussed in Section 3.2, evaluation of the cross-derivative terms may require small time-step durations with a highly skewed grid. To study the effect of calculating the cross-derivative terms with a partially implicit method, the proposed finite-difference model (FDM) and a fully implicit FDM (FI-FDM) are used. The FI-FDM is identical to the FDM presented in this paper, except for the treatment of the cross-derivative terms. The preconditioned biconjugate gradient method permits the system matrix to be non-diagonally dominant. By effective preconditioning, the FI-FDM can be converged on a 19-point stencil. HYDRUS is also run for comparison. As discussed in Section 4.1, the use of a highly skewed mesh may lead to a decrease in the accuracy of the results. To assess the effect of grid skewness on the accuracy of the proposed model, four different types of grids with a 0.2 × 1 m rectangular domain are considered, as shown in Fig. 7. To estimate the numerical accuracy, grids of 20 × 20 mesh, 20 × 40 mesh, and 20 × 80 mesh are used. Grid 1 is orthogonal. The minimum

grid angles for grids 2, 3, and 4 are 50°, 30°, and 20°, respectively. Because HYDRUS uses a unstructured triangular grid, HYDRUS grids are generated by modifying the structured FDM grid. The nodes of both the grids are arranged such that they are located at the same coordinate positions. The loam soil properties listed in Table 1 are considered. The top and side boundaries are no-flow boundaries, while the bottom boundary is $\psi_{bot} = 0.5$ m. The initial condition is $\psi_{init} = -1$ m, and the simulation period is 1 day. The time-step duration is automatically adjusted by the algorithm presented earlier.

Fig. 8 shows the pressure head profiles for $x = 0.1$ m at the end of the simulation carried out using the three models. In this test simulation, the dense orthogonal grid ($\Delta z = 0.0025$ m) solution was considered the surrogate solution for the exact solution. The solutions obtained by the three models approached the exact solution with an increase in the grid fineness in the case of Grids 1 and 2 but not in the case of Grids 3 and 4. Table 2 lists the relative errors in the pressure head and the mass balance errors generated by the three models. The FDM and FI-FDM gave relative errors with the same order in all cases, indicating that the numerical accuracy of the test simulation was unaffected by the method used to evaluate the cross-derivative terms. The order of the errors obtained with HYDRUS was almost the same as that of the errors obtained with the FDM and FI-FDM. The FDM has a first-order temporal and a second-order spatial precision in the computational space. HYDRUS also has a first-order temporal and a second-order spatial precision in the physical space. When Grid 1, i.e., the orthogonal grid, was used, the relative errors in the three models decreased with an increase in the spatial resolution of the grid. However, when Grids 3 and 4 were used, the relative errors in the three models were larger than the relative errors obtained with Grid 1, and the errors did not

decrease with an increase in the grid fineness. These results suggested that the model accuracy decreased when a highly skewed mesh was used. The FDM and FI-FDM showed virtually perfect mass conservation in this test simulation, whereas HYDRUS generated comparatively large mass balance errors, especially when Grids 2 and 4, with dimensions of 20×80, were used. The mass balance performances of the three models are shown in Fig. 9. Even though the FDM and FI-FDM underestimated the total amount of additional mass and net flux when Grid 4 was used, their balances were virtually perfect. On the other hand, HYDRUS apparently overestimated the net flux flowing into the domain, because of which mass balance errors were observed.

Table 3 lists the CPU time and the total number of iterations required for the three models. When using Grids 3 and 4, the number of iterations and CPU time required for the FDM were greater than those required by the FI-FDM whereas there was no significant difference in the total iteration number and the CPU time consumed when using Grid 1 and 2. When Grids 3 and 4 were used, it was found that the implicit evaluation of the cross-derivative terms of the FI-FDM permitted large time-step durations. On the other hand, the FDM consumed fewer CPU resources per iteration than the FI-FDM, because very few unknown values were included in the simultaneous equation of the former, as mentioned in Section 3.2. Hence, when the contribution by the cross-derivative terms is large, we must make a choice between the FDM, which uses fewer CPU resources per iteration, and the FI-FDM, which requires a small number of iterations. There was no significant difference in the total CPU time consumed by the FDM and FI-FDM in this test even when a highly skewed mesh was used. However, for very difficult problems (e.g., problems with a high degree of heterogeneity and anisotropy with highly skewed grids),

explicit evaluation of the cross-derivative terms could significantly affect the convergence speed and accuracy. In all cases, HYDRUS required a greater number of iterations than FDM and FI-FDM. It was not possible to compare the CPU time consumed by the FDM and that consumed by the HYDRUS because the two models were run on different operating systems and were coded by different programming languages. Hence, the CPU time consumed by HYDRUS is shown only for reference.

This test simulation was carried out to estimate the effect of grid skewness on the model performance and accuracy. A comparison between the FDM and the FI-FDM made by considering all the terms fully implicitly showed that the proposed method for evaluating the cross-derivative terms did not cause any decrease in the numerical accuracy in this test case; further, small time-step durations with highly skewed grids were required. However, in certain other cases, there was a possibility of the numerical accuracy being affected, as mentioned above. We also confirmed that the proposed model and the finite-element model generally had the same numerical accuracy. Furthermore, the mass conservation behavior of the proposed model was better than that of the finite-element model in this simulation, considering that these two models used the same mixed-form of Richards' equation.

4.3. Test 3: transient variably saturated flow in two dimensions

The model performance for a non-rectangular domain is tested in this simulation. A trapezoidal domain shown in Fig. 10 is considered, and the sandy loam soil properties listed in Table 1are used. No-flow conditions are imposed at the top and bottom boundaries. The left-side boundary is given as $\psi_{ls} = 1.5 - z$ m where $z \leq 1.5$ m, and a no-flow boundary is considered, where 1.5 m $< z$. The right-side boundary is given as $\psi_{rs} = 0.5 - z$

m, where $z \leq 0.5$ m, and a seepage-face boundary, where 0.5 m $< z$, is considered. The initial condition is $\psi_0 = 0.5 - z$ m, and the simulation period is 1 week. The time-step duration is adjusted automatically as $1 \leq \Delta t \leq 60$ min. Fig. 11 shows the grids (30×30) used for the FDM and HYDRUS. The node positions in both the grids are set to be identical.

Fig. 12 shows the ψ contours simulated by the FDM and HYDRUS. The two results are in good agreement and successfully describe water flows in the soil. Table 4 lists the relative error and model performance determined in Test 3. In this test simulation, a dense-grid (150×150) solution is used as a surrogate solution for the exact solution. As in the case of Test 2, the proposed model and HYDRUS give relative errors of the same order. The number of iterations required by the proposed model is less than that required by HYDRUS. The two models are run on the same computer; however, the FDM is coded by C++ and operated on the Linux system, whereas HYDRUS is coded by Fortran and operated on the Windows system. The two models also use different types of grids. It is not possible to assess the model efficiency on the basis of the CPU time required for calculation. However, when the node number and node positions used are the same, the FDM converges more quickly than HYDRUS; this is because the total number of iterations required for calculation is dependent on the numerical algorithm but independent of the programming languages and operating system used. This can be confirmed from Fig. 13,which shows the time-step durations of the two models for this calculation. The time-step durations of both the models are controlled by the same rule and adjusted on the basis of the number of iterations required for convergence at the previous time step. As shown in Fig. 13, the FDM converges more rapidly than HYDRUS in this simulation.

4.4. Test 4: Simulation of rainfall-runoff on a slope in three dimensions

In Test 4, the FDM and conventional FDM, which use high-resolution stepwise orthogonal grids, are compared in a three-dimensional curvilinear domain. A rainfall-runoff simulation for a slope is carried out. There are many unsolved problems associated with the simulation of rainfall runoff for a real slope, including insufficient knowledge of detailed runoff processes, slope heterogeneity, and high computational cost. However, even with these problems, the numerical approach can be a useful tool for studying the runoff processes of a slope (e.g., *Kosugi et al.* [2004]; *Keim et al.* [2006]; *Tani* [2008]; *An et al.* [2008]; *Liang et al.* [2009]).

A curvilinear slope, as illustrated in Fig. 14, was considered, and the sandy soil properties listed in Table 1 were used. The mesh shapes for the FDM, HYDRUS, and conventional FDM simulations are shown in Fig. 15. The number of nodes for the FDM, HYDRUS, and conventional FDM were 4896, 4560, and 34441, respectively. No-flow conditions were imposed at the top, bottom, and side boundaries. The surface boundary was a flux boundary, and the bottom boundary was a seepage-face boundary. The initial conditions were $\psi = 0$ at the intersection of the bottom and lower nodes and $\psi = -1$ m on the intersection of the surface and the upper nodes; the values for the other nodes were between those of the top and bottom nodes and were linear functions of z. The rainfall intensity is as shown in Fig. 16. The simulation period was 2 weeks, and the time-step duration was $1 \leq \Delta t \leq 3600$ s.

Fig. 17 shows the pressure head results obtained with the FDM, HYDRUS, and the conventional FDM. The discharge flow rates and cumulative discharge volumes at the lower end of the slope are illustrated in Fig. 18. The three results are in good agreement with

one another. The FDM can be used for the successful simulation of the three-dimensional flow on a curvilinear slope, despite the use of a low-resolution grid. The conventional FDM also appears to be suitable for a successful simulation of the test problem. However, for a successful simulation of the test problem, the conventional FDM must have a high-resolution grid, as shown in Fig. 15. If a low-resolution grid is used in the conventional FDM, poor results will be obtained because of the insufficient representation of the curvilinear domain. Table 5 lists the statistics of the model performance. Owing to additional resolution, the conventional FDM consumes more CPU time than the FDM. If the domain complexity increases, the additional computational costs may increase because of the higher-resolution required. This implies that when using the FDM, the computational time can be considerably reduced, as compared to the case where the conventional FDM is used, as in the case of iterative parameter estimation or the Monte Carlo exercise, which usually requires hundreds or thousands of runs to arrive at an ideal parameter set or an objective function. Fig. 19 shows the time-step durations. As shown in this figure, the FDM and conventional FDM converge at almost identical speeds, although a higher-resolution grid is used in the latter. We believe that the partially explicit evaluation of the cross-derivative terms of the FDM affects the convergence speed and that the convergence speed of the FDM becomes almost equal to that of the conventional FDM.

5. Conclusion

In this paper, we present a finite-difference saturated-unsaturated flow model that can fit a curvilinear flow domain. A coordinate transformation method is applied, and hence, the model can be used to handle complex geometries and anisotropies. The proposed scheme has first-order temporal and second-order spatial discretization precisions. The proposed

scheme has several practical advantages in terms of ease of coding and low computation storage; this is because the cross-derivative terms are evaluated at the previous iteration step, and the linearized equation uses a 7-point stencil, as in the case of the conventional finite-different model on an orthogonal grid. Test simulations are carried out to examine the effect of the method used for evaluating the cross-derivative terms.

Four simulations were carried out to assess the performance of the proposed model. In the first test simulation, a two-dimensional steady-state condition with a highly skewed grid was simulated. The isolines of the pressure head produced by the proposed model were insensitive to the grid shape, and the results were in good agreement with the correct solution. The numerical accuracy decreased when a highly skewed grid was used. The second test was a two-dimensional unsteady-state simulation; here, the mesh skewness and mesh size were varied. To evaluate the effect of the manner in which cross-derivative terms were treated, simulation was carried out with the FI-FDM. Comparison of the accuracy and CPU time of the proposed model with those of the FI-FDM showed that the method of treating the cross-derivative terms did not affect the numerical accuracy in this test case. However, the FI-FDM was faster than the proposed model in which a highly skewed mesh was used, whereas the proposed model was faster than the FI-FDM in which a mesh that was not highly skewed was used. For simulating extremely complex domains, the FI-FDM might be more preferable than the proposed model. The finite-element model with an unstructured triangular grid was also used in the second, third, and fourth test simulations, in order to compare the performance of the finite-element model with that of the proposed model. In the second test simulation, the proposed model converged more rapidly than the finite-element model; however, the relative errors

in both the models were of the same order when the node positions were identical. A two-dimensional transient, variably saturated flow with a trapezoidal flow domain was simulated in the third test simulation. The proposed model delivered a good performance with the non-rectangular flow domain and converged more rapidly than the finite-element model; both the models gave the same order of relative error. The fourth test involved a three-dimensional simulation of rainfall run-off on a curvilinear slope. The proposed model could successfully reproduce the three-dimensional subsurface flow, and the model results agreed well with those obtained using the finite-element model and the conventional FDM, in which higher-resolution orthogonal grids were used. Thus, we concluded that the proposed model is more efficient than the conventional FDM because it can be used to represent a curvilinear shape even when employing a comparatively low resolution grid. As seen from the second and third test simulations, the proposed model converged more rapidly than the finite element model. Overall, in the test simulations, the proposed model gave a smaller mass balance error than the finite-element model; further, the proposed model converged faster than the finite-element model. Both the models gave relative errors of the same order.

Appendix: Derivation of Eq. (3)

A chain rule operation was performed to transform a partial derivative of a function f in physical space (x_1, x_2, x_3) into that in computational space (ξ_1, ξ_2, ξ_3) as follows:

$$\frac{\partial f}{\partial x_r} = \sum_{p=1}^{3} \frac{\partial \xi_p}{\partial x_r} \frac{\partial f}{\partial \xi_p} \tag{24}$$

or

$$\frac{\partial f}{\partial x_r} = \sum_{p=1}^{3} J \frac{\partial}{\partial \xi_p} \left(\frac{1}{J} \frac{\partial \xi_p}{\partial x_r} f \right). \tag{25}$$

Applying Eq. (25) to Eq. (2) gives

$$\frac{\partial\theta}{\partial t} = \sum_{r=1}^{3}\sum_{s=1}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left\{\frac{K_{r,s}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\psi}{\partial x_s}\right\} + \sum_{r=1}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left(\frac{K_{r,3}}{J}\frac{\partial\xi_p}{\partial x_3}\right). \tag{26}$$

Application of Eq. (24) to $\partial\psi/\partial x_r$ of Eq. (26) gives

$$\frac{\partial\theta}{\partial t} = \sum_{r=1}^{3}\sum_{s=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} J\frac{\partial}{\partial\xi_p}\left\{\frac{K_{r,s}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\xi_q}{\partial x_s}\frac{\partial\psi}{\partial\xi_q}\right\} + \sum_{r=1}^{3}\sum_{p=1}^{3} J\frac{\partial}{\partial\xi_p}\left(\frac{K_{r,3}}{J}\frac{\partial\xi_p}{\partial x_3}\right). \tag{27}$$

This equation can be rewritten as

$$\frac{1}{J}\frac{\partial\theta}{\partial t} = \sum_{p=1}^{3}\sum_{q=1}^{3}\frac{\partial}{\partial\xi_p}\left\{G^{p,q}K\frac{\partial\psi}{\partial\xi_q}\right\} + \sum_{p=1}^{3}\frac{\partial}{\partial\xi_p}\left(H^pK\right), \tag{28}$$

where

$$J = \frac{\partial(\xi_1,\xi_2,\xi_3)}{\partial(x_1,x_2,x_3)}, \qquad G^{p,q}K = \sum_{r=1}^{3}\sum_{s=1}^{3}\frac{K_{r,s}}{J}\frac{\partial\xi_p}{\partial x_r}\frac{\partial\xi_q}{\partial x_s}, \qquad H^pK = \sum_{r=1}^{3}\frac{K_{r,3}}{J}\frac{\partial\xi_p}{\partial x_3}. \tag{29}$$

Acknowledgments. This work was supported by a grant-in-aid for the Japan Society for the Promotion of Science (JSPS) fellows and by a grant-in-aid for scientific research (A, 20246082) from the JSPS.

References

An, H., Ichikawa, Y., Tachikawa, Y., and Shiiba, M. (2008), Developing a three-dimensional coupled model of pipe-matrix subsurface flow, *Hydrological Research Letters of Japan Society of Hydrology and Water Resources, 2*, 52-55.

Bause, M., Knabner, P., and Bergamaschi, L. (2004), Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods, *Advances in Water Resources, 27*, 145-144.

Bergamaschi, L. and Putti, M. (1999), Mixed finite elements and Newton-type linearizations for the solution for the unsaturated flow equation, *International Journal for Numerical Methods in Engineering, 45*, 1025-1046.

Billstein, M., Svensson, U. and Johansson, N. (1999), Development and validation of a numerical model of flow through embarkment dams - comparisons with experimental data and analytical solutions, *Transport in Porous Media, 35*, 395-406.

Broadbridge, P. and White, I. (1988), Constant rate rainfall infiltration: A versatile nonlinear model, 1, *Analytic solution, Water Resources Research, 24*, 145-154.

Carsel, R.F. and Parrish, R.S. (1988), Developing joint probability distributions of soil water retention characteristics, *Water Resources Research, 24*, 755-769.

Celia, M.A., Bouloutas, E.T., and Zarba, R.L. (1990), A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resources Research, 26(7)*, 1483-1496.

Chapman, T.G. and Ong, G. (2006), A new equation for shallow groundwater flow over a curved impermeable boundary: Numerical solutions and laboratory tests, *Water Resources Research, 42*, W03427, doi:10.1029/2005WR004437.

Chen, X. (2004), A Cartesian method for fitting the bathymetry and tracking the dynamic position of the shoreline in a three-dimensional, hydrodynamic model, *Journal of computational physics, 200*, 749-768.

Chung, T.J. (2002), Computational fluid dynamics. Cambridge University Press.

Clement, T.P., Wise, W. R., and Molz, F.J. (1994), A physically based, two-dimensional, finite-difference algorithm for modeling variably saturated flow, *Journal of Hydrology, 161*, 69-91.

Cooley, R.L. (1971), A finite difference method for unsteady flow in variably saturated porous media: application to a single pumping well, *Water Resources Research, 7*, 1607-1625.

Dane, J.H. and Mathis, F.H. (1981), An adaptive finite difference scheme for the one-dimensional water flow equation, *Soil Science Society of America, 45*, 1048-1054.

Dogan, A. and Motz, L.H. (2005), Saturated-unsaturated 3D groundwater model. I: development, *Journal of Hydrologic Engineering, 10(6)*, 492-504.

Farthing, M.W., Kees, C.E., and Miller, C.A. (2003), Mixed finite element methods and higher order temporal approximations for variably saturated groundwater flow, *Advances in Water Resources, 25*, 85-101.

Fassino, C. and Manzini, G. (1999), Fast-secant algorithms for the non-linear Richards equation, *Communications in Numerical Methods in Engineering, 14*, 921-930.

Freeze, R.A. (1971), Three dimensional transient, saturated-unsaturated flow in a groundwater basin, *Water Resources Research, 7*, 153-171.

Forsyth, P.A., Wu, Y.S., and Pruess, K. (1995), Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media, *Advances in Water Resources, 18*, 25-38.

Haverkamp, R. and Vauclin, M. (1981), A comparative study of three forms of the Richard equation used for predicting one-dimensional infiltration in unsaturated soil, *Soil Science Society of America, 45*, 13-20.

Hogarth, W.L., Parlange, J.Y. (2000), Application and improvement of a recent approximate analytical solution of Richards' equation, *Water Resources Research, 36*, 1965-1968.

Hsu, M., Teng, W., and Lai, C. (1998) Numerical simulation of supercritical shock wave in channel contraction, *Computers & Fluids, 27*, 347-365.

578 Huang, K., Zhang, R., and Van Genuchten, M. (1994), Eulerian-Lagrangian approach
579 with an adaptively corrected method of characteristics to simulate variably saturated
580 flow, *Water Resources Research, 30*, 499-507.

581 Huang, K., Mohantry, B.P., and Van Genuchten, M. (1996), A new convergence criterion
582 for the modified Picard iteration method to solve the variably saturated flow equation,
583 *Journal of Hydrology, 178*, 69-91.

584 Jie, Y., Jie, G., Mao, Z. and Li, G. (2004), Seepage analysis based on boundary-fitted
585 coordinate transformation method, *Computers and Geotechnics, 31*, 279-283.

586 Jones, J.E. and Woodward C.S. (2001), Newton-Krylov-Multigrid solvers for large-scale,
587 highly heterogeneous, variably saturated flow problems, *Advances in Water Resources,*
588 *24*, 763-774.

589 Kavetski, D., Binning, P., and Sloan, S.W. (2002), Noniterative time stepping schemes
590 with adaptive truncation error control for the solution of Richards equation, *Water*
591 *Resources Research, 38(10)*, 1211.

592 Keim, R.F., Tromp-van Meerveld H.J., and McDonnell, J.J. (2006), A virtual experi-
593 ment on the effects of evaporation and intensity smoothing by canopy interception on
594 subsurface stormflow generation. *Journal of Hydrology, 327*, 352-364.

595 Kershaw D.S. (1981), Differencing of the diffusion equation in Lagrangian hydrodynamic
596 codes. *Journal of Computational Physics, 39*, 375-95.

597 Kinouchi, T., Kanda, M. and Hino, M. (1991), Numerical simulation of infiltration and
598 solute transport in an s-shaped model basin by a boundary-fitted grid system, *Journal*
599 *of Hydrology, 122*, 373-406.

Kogaki, T. (1999), Finite difference schemes in generalized coordinates and their application to large eddy simulation. Ph.D. thesis, University of Tokyo. (In Japanese)

Koo, M. and Leap, D.I. (1998), Modeling three-dimensional groundwater flows by the body-fitted coordinate (BCF) method: I. stationary boundary problems, *Transport in Porous Media, 30*, 217-239.

Koo, M. and Leap, D.I. (1998), Modeling three-dimensional groundwater flows by the body-fitted coordinate (BCF) method: II. free and moving boundary problems, *Transport in Porous Media, 30*, 345-362.

Kosugi, K., Uchida, T., and Mizuyama, T. (2004), Numerical calculation of soil pipe flow and its effect on water dynamics in a slope, *Hydrological Processes, 18*, 777-789.

Kotakemori, H., Hasegawa, H., and Nishida, A. (2005), Performance evaluation of a parallel iterative method library using OpenMP, *In Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region,* 432-436, Beijing, November, 2005.

Lakner, M., Plazl, I. (2008), The finite differeces method for solving systems on irregular shapes, *Computers and Chemical Engineering, 32*, 2891-2896.

Lehmann, F. and Ackerer, Ph. (1998), Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media, *Transport in Porous Media, 31*, 275-292.

Liang, W., Kosugi, K., and Mizuyama, T. (2009), A three-dimensional model of the effect of stemflow on soil water dynamics around a tree on a hillslope, *Journal of Hydrology, 366*, 62-75.

Maliska, C.R. and Raithby, G.D. (1984), A method for computing three dimensional flows using non-orthogonal boundary-fitted coordinates, *International Journal for Numerical Methods in Fluids, 4*, 519-537.

Mansell, R.S., Liwang, Ma., Ahuja, L.R., and Bloom, S.A. (2002), Adaptive grid refinement in numerical models for water flow and chemical transport in soil: a review, *Vadose Zone, 1*, 222-238.

Manzini, G., and Ferraris, S. (2004), Mass-conservative finite volume methods on 2-D unstructured grids for the Richards' equation, *Advances in Water Resources, 27*, 1199-1215.

Mastin, C.W. (1998), Truncation error on structured grids. In: Tomson J.F., Soni B.K., Weatherill N.P., editors. Handbook of Grid Generation. CRC Press LLC, Florida, USA, 32-1–10.

Menziani, M., Pugnaghi, S., and Vincenzi, S. (2007), Analytical solutions of the linearized Richards equation for discrete arbitrary initial and boundary conditions, *Journal of Hydrology, 332*, 214-225.

Morinishi, Y., Lund, T.S., Vasilyev, O.V., and Moin, P. (1998), Fully conservative higher order finite difference schemes for incompressible flow, *Journal of computational physics, 143*, 90-124.

Mualem, Y. (1976), A new model for predicting hydraulic conductivity of unsaturated porous media, *Water Resources Research, 12*, 513-522.

Murray, R.J. and Reason C.J.C. (2001), A curvilinear version of the Bryan-Cox-Semtner Ocean Model and Its Representation of the Artic Circulation, *Journal of computational physics, 171*, 1-46.

Neuman, S.P. (1973), Saturated-unsaturated seepage by finite elements, *Journal of Hydraulics Division, ASCE,* HY12, 2233-2250.

Paniconi, C., Aldama, A.A., and Wood, E.F. (1991), Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation, *Water Resources Research, 27(6)*, 1147-1163.

Paniconi, C. and Putti, M. (1994), A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, *Water Resources Research, 30(12)*, 3357-3374.

Parlange, J. (1972), Theory of water movement in soils, 6, Effect of water depth over soil, *Soil Science Society of America, 113*, 308-312.

Peric, M. (1994), A finite volume method for the prediction of three-dimensional fluid flow in complex ducts, Ph.D. thesis, University of London, Imperical College.

Philip, J.R. (1957), The theory of infiltration, 1, The infiltration equation and its solution, *Soil Science Society of America, 83*, 345-357.

Ruhaak, W., Rath, V., Wolf, A. and Clauser, C. (2008), 3D finite volume groundwater and heat transport modeling with non-orthogonal grids, using a coordinate transformation method, *Advances in Water Resources, 31*, 513-524.

Šimůnek, J., Sejna, M., Van Genuchten, M.T. (1999), HYDRUS-2D:simulating water flow and solute transport in two-dimensional variably saturated media, Tech, Res., IGWMC, Golden, CO, USA.

Tani, M. (2008), Analysis of runoff-storage relationships to evaluate the runoff-buffering potential of a sloping permeable domain, *Journal of Hydrology, 360*, 132-146.

667 Thompson, J.F., Soni, B.K., and Weatherill, N.P. (1999), Handbook of grid generation,
668 USA: CRC Press LLC.

669 Tocci, M.D., Kelley, C.T., and Miller, C.T. (1997), Accurate and economical solution of
670 the pressure-head form of Richards' equation by the method of lines. *Advances in Water*
671 *Resources, 20*, 1–14.

672 Van Genuchten, M. A. (1980), Closed-form equation for predicting the hydraulic conduc-
673 tivity of unsaturated soils, *Soil Science Society of America, 44*, 892-898.

674 Weeks, S.W., Sander, G.C., Braddock, R.D. and Matthews, C.J. (2004), Saturated and
675 unsaturated water flow in inclined porous media, *Environmental Modeling and Assess-*
676 *ment, 9*, 91-102.

677 Wesseling, P. (2001), Principle of computational fluid dynamics, Berlin: Springer.

678 Zang, Y., Street, R.L., and Koseff, J.R. (1994), A non-staggered grid, fractional step
679 method for time-dependent incompressible Navier-Stokes equations in curvilinear coor-
680 dinate, *Journal of Computational Physics, 114*, 18-33.

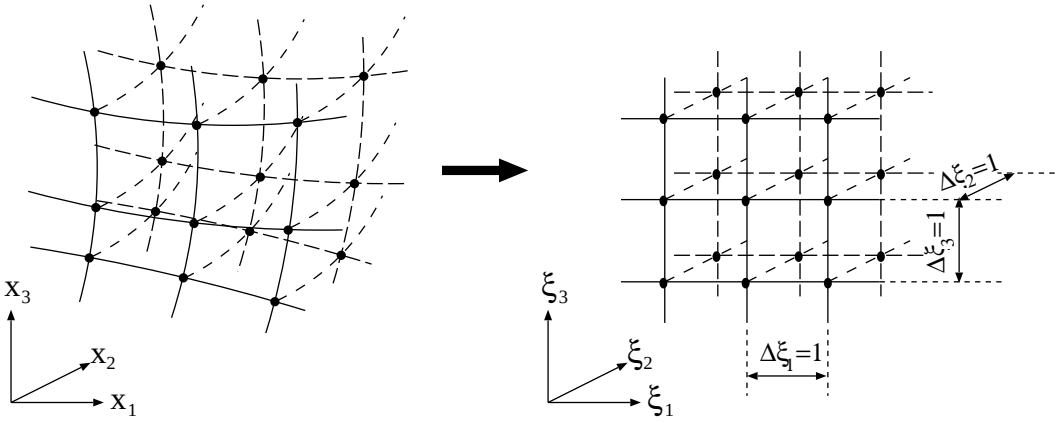

Figure 1. Concept of coordinate transformation: an arbitrarily shaped mesh in physical space is transformed into an orthogonal mesh in computational space.

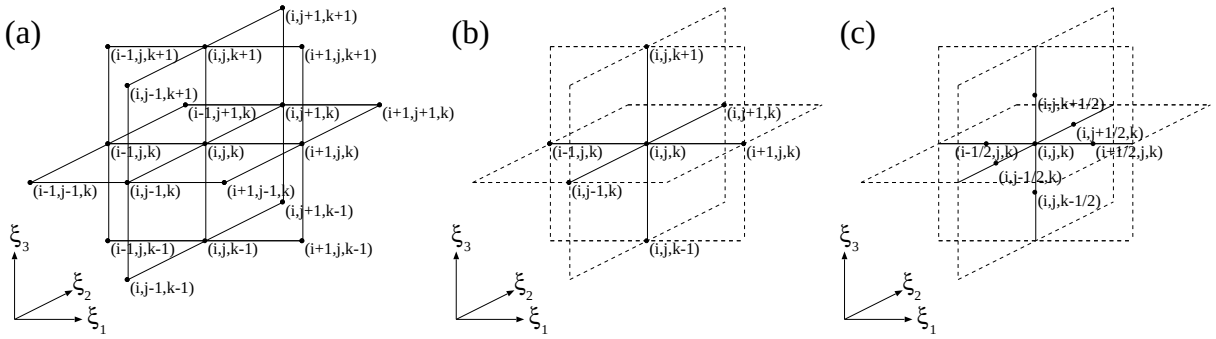

Figure 2. Evaluated (a) 19-point stencil of ψ and (b) 7-point stencil of ψ. (c) Utilized 7-point stencil of K and G in an iterative procedure.

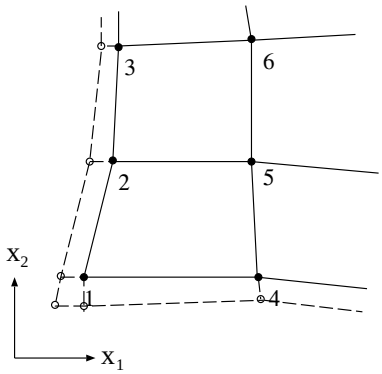

Figure 3. Additional row of ghost nodes (unfilled circle) with zero thickness for evaluating the metrics at boundary nodes.

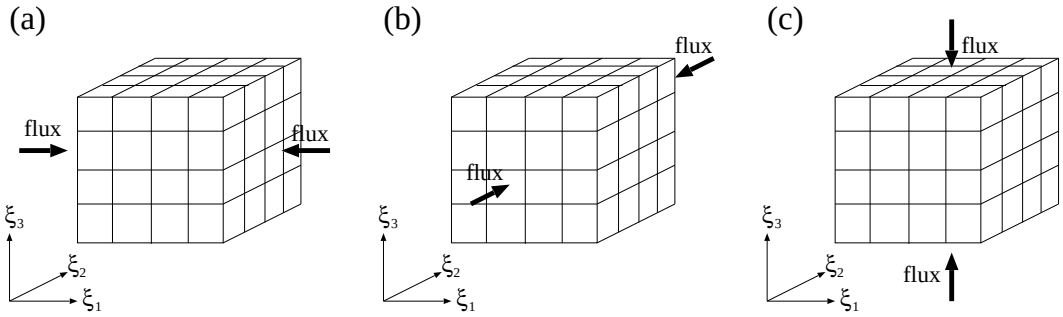

Figure 4. Flux directions under various boundary conditions.

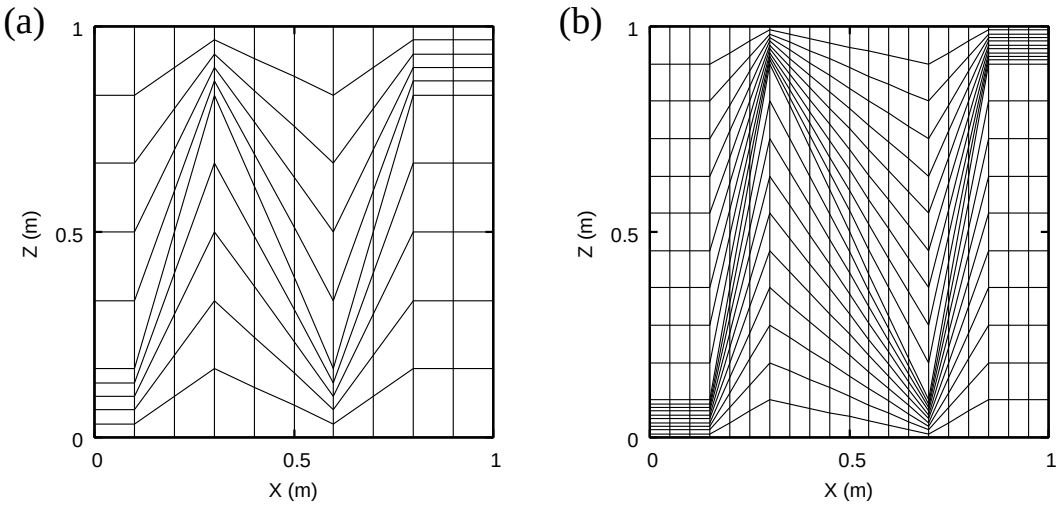

Figure 5. (a) The 10 × 10 and (b) 20 × 20 Kershaw meshes used in Test 1.

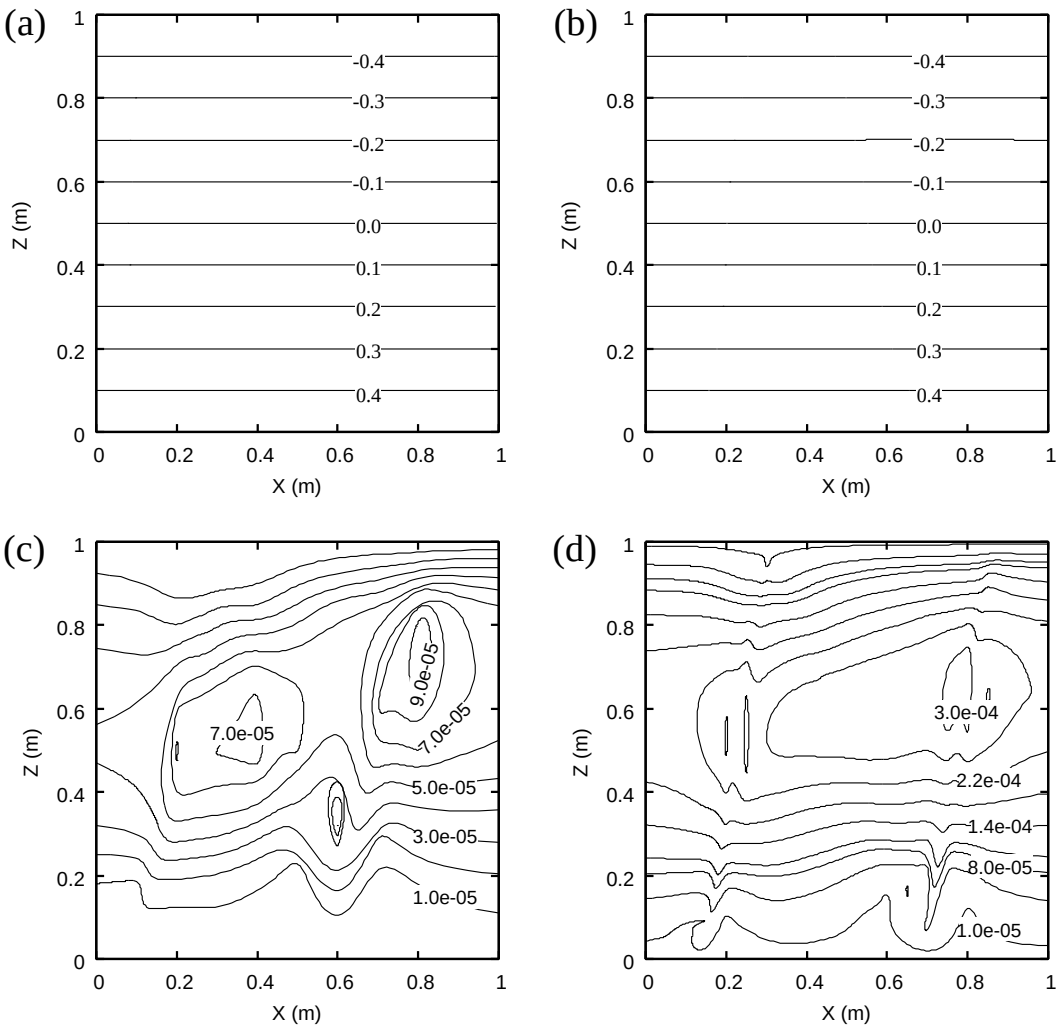

Figure 6. Isolines of pressure head with (a) 10 × 10 mesh and (b) 20 × 20 mesh and the root mean square error of pressure head for the (c) 10 × 10 mesh and (d) 20 × 20 mesh results in Test 1.

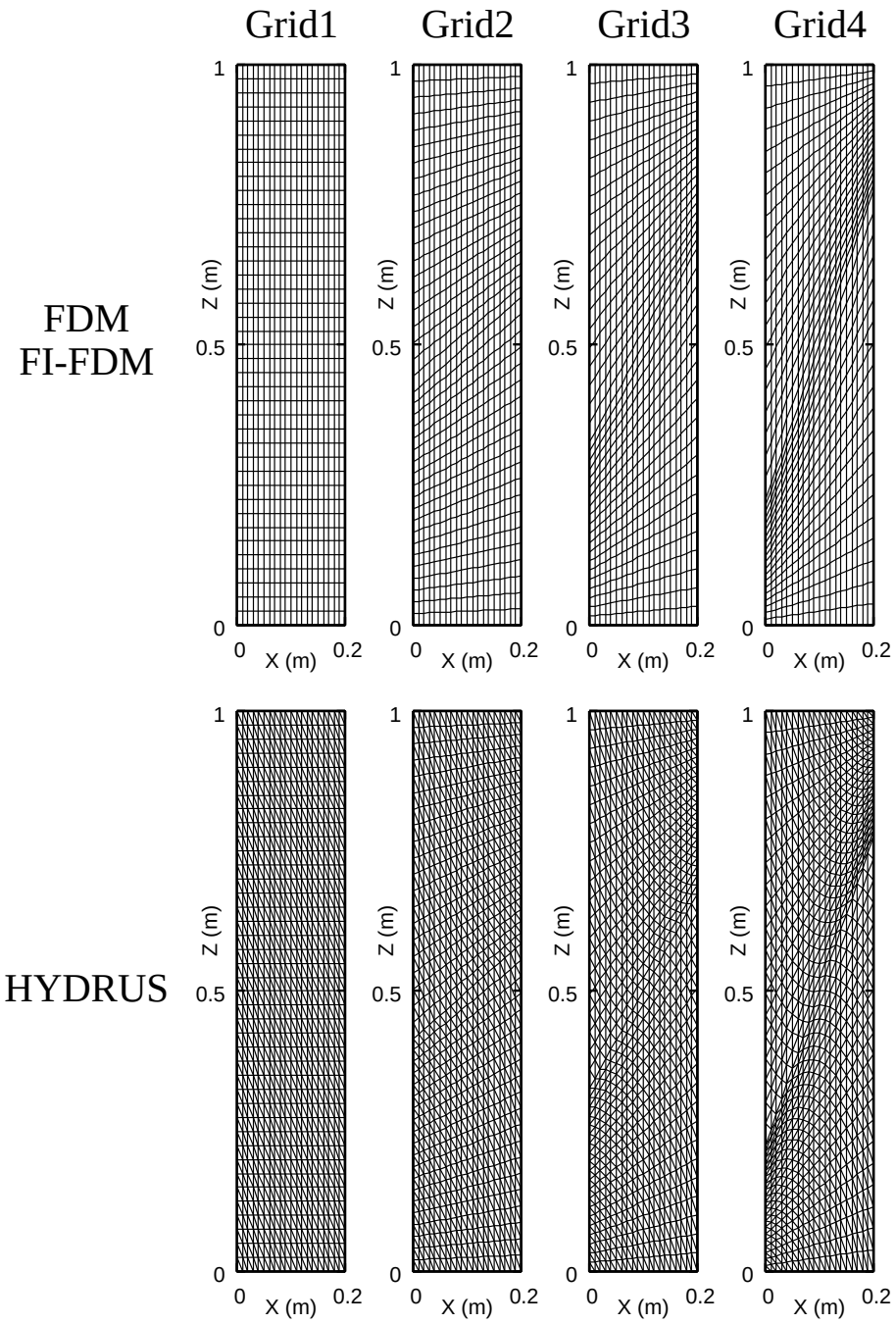

Figure 7. Four types of grids used in Test 2 (20×40).

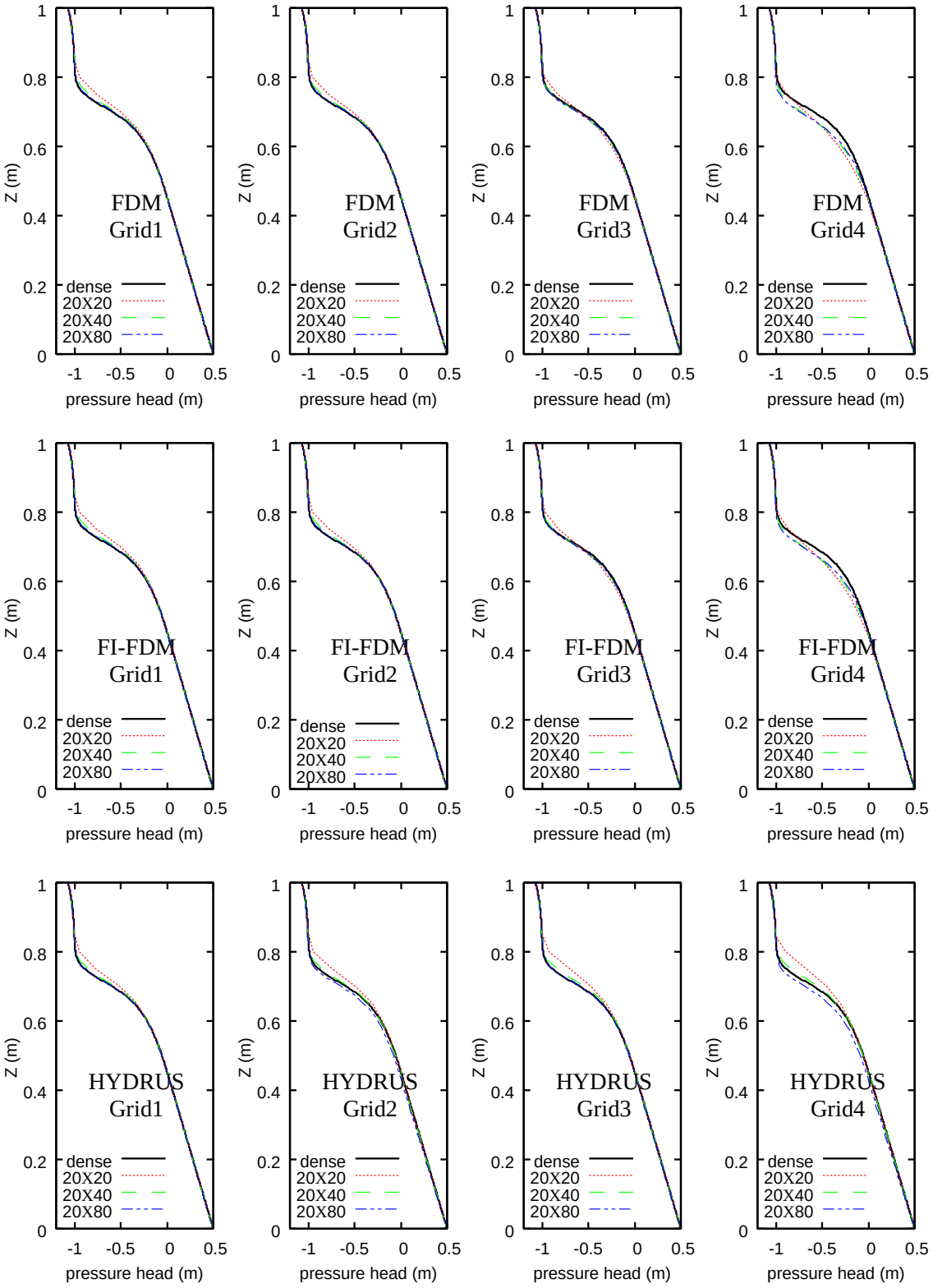

Figure 8. Pressure head profiles on $x = 0.1$ m obtained at the end of simulations carried out by FDM, FI-FDM, and HYDRUS in Test 2.

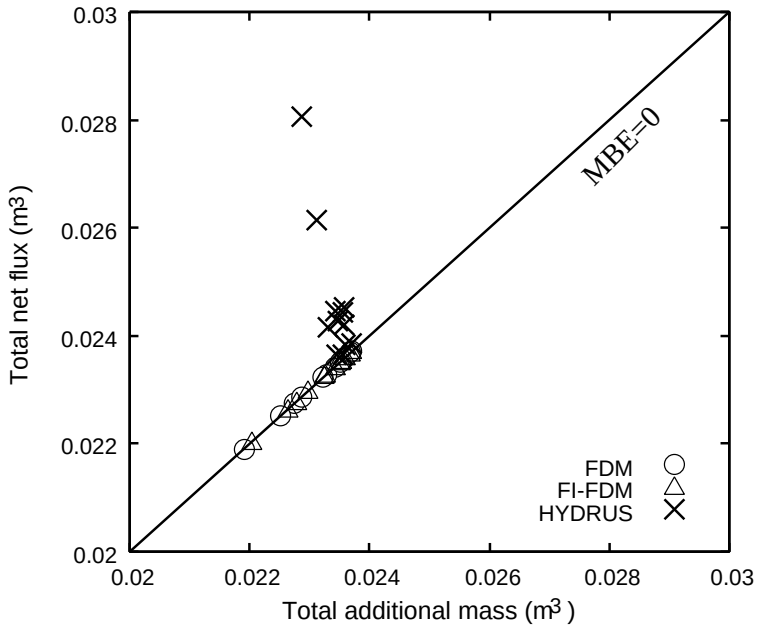

Figure 9. Mass balances of FDM, FI-FDM, and HYDRUS in Test 2.

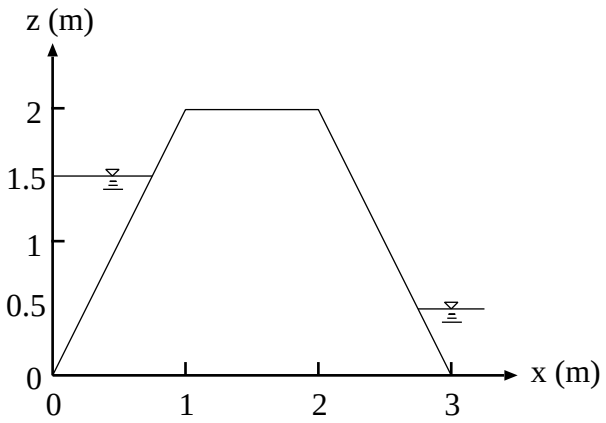

Figure 10. Flow domain in Test 3.

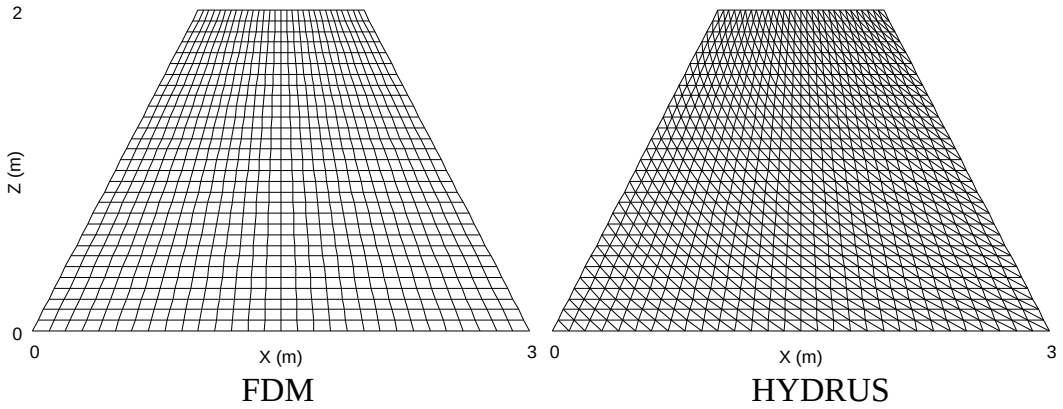

Figure 11. Grids used in Test 3.

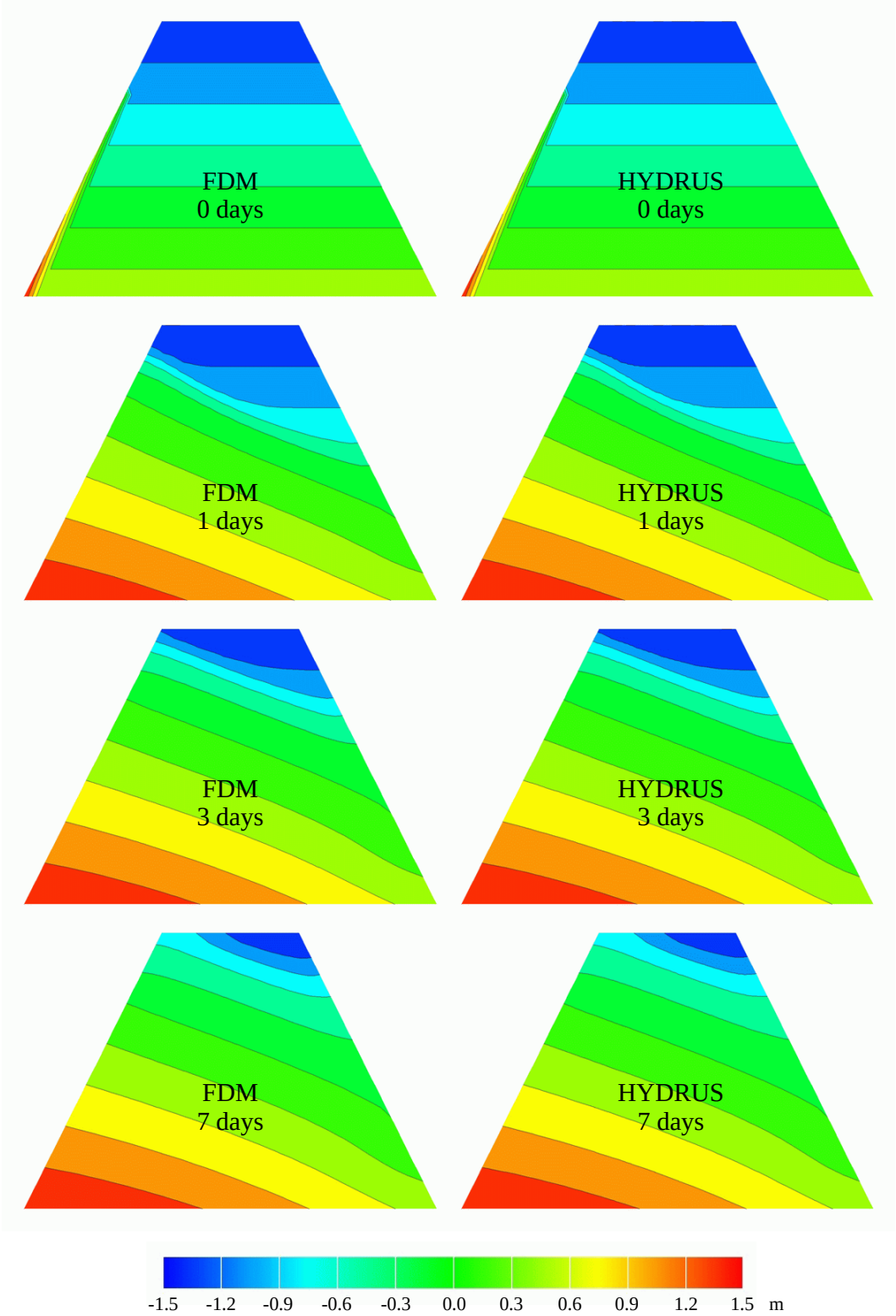

Figure 12. Contours of pressure head simulated in Test 3; the left and right sides show the results obtained by FDM and HYDRUS, respectively.

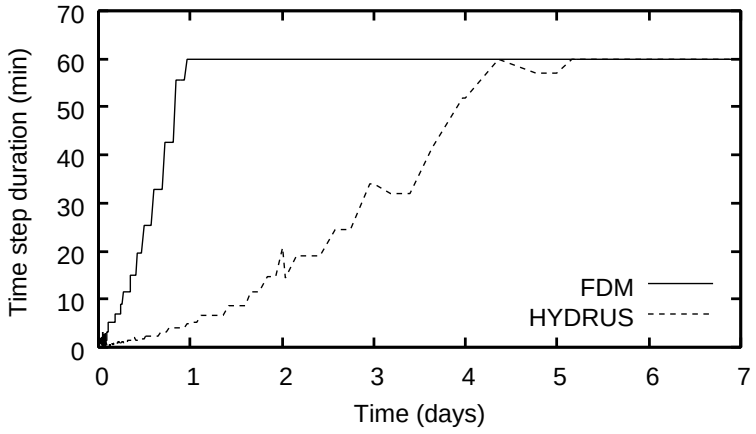

Figure 13. Time-step durations of FDM and HYDRUS in Test 3.

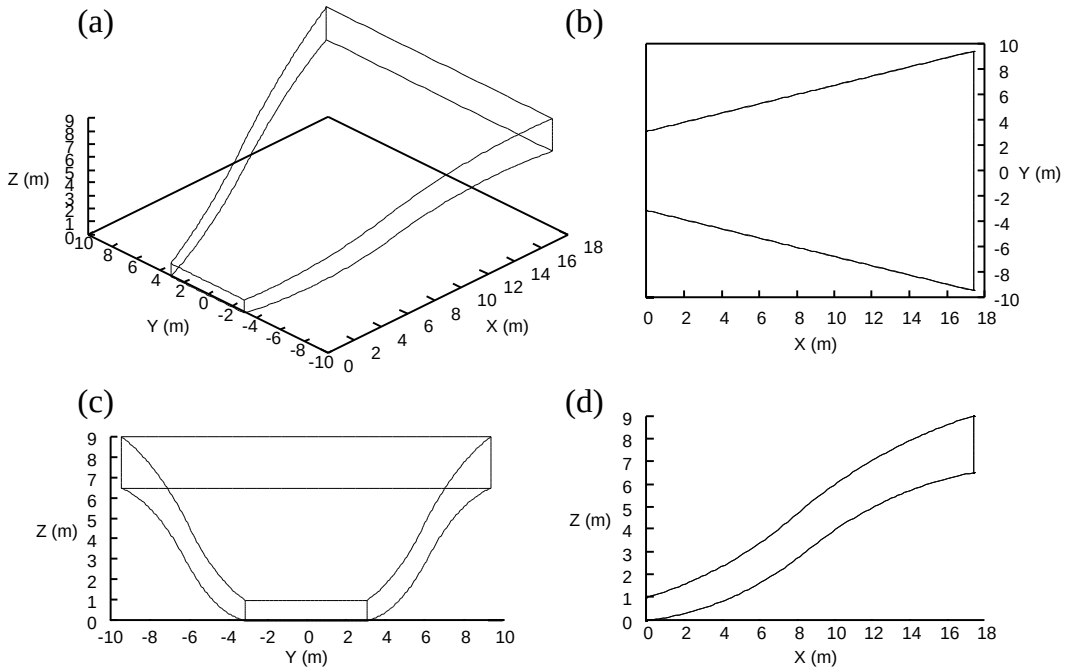

Figure 14. An inclined domain used in Test 4. (a) Perspective view; (b) plan view; (c) vertical cross section from x,y coordinate (0,0); (d) vertical section from x,y coordinate (9,-10).

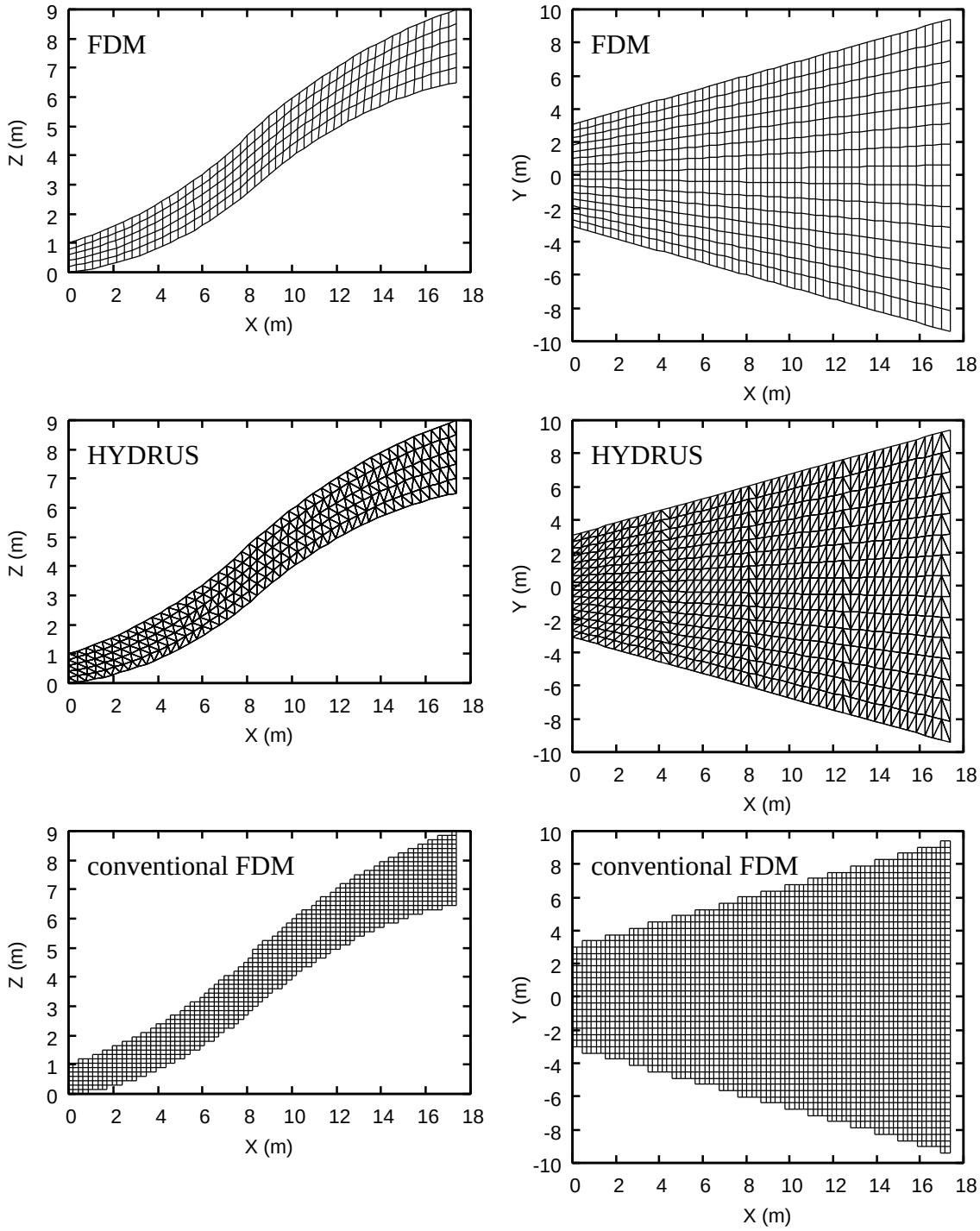

Figure 15. Grids used in Test 4; left side describes vertical sections from x,y coordinate (9,-10) and right side gives plan views.

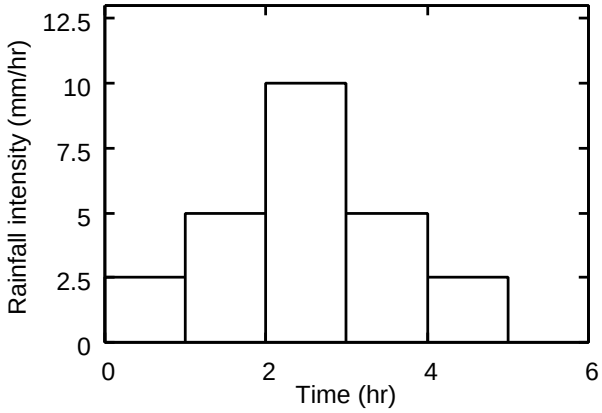

Figure 16. Rainfall intensity for Test 4.

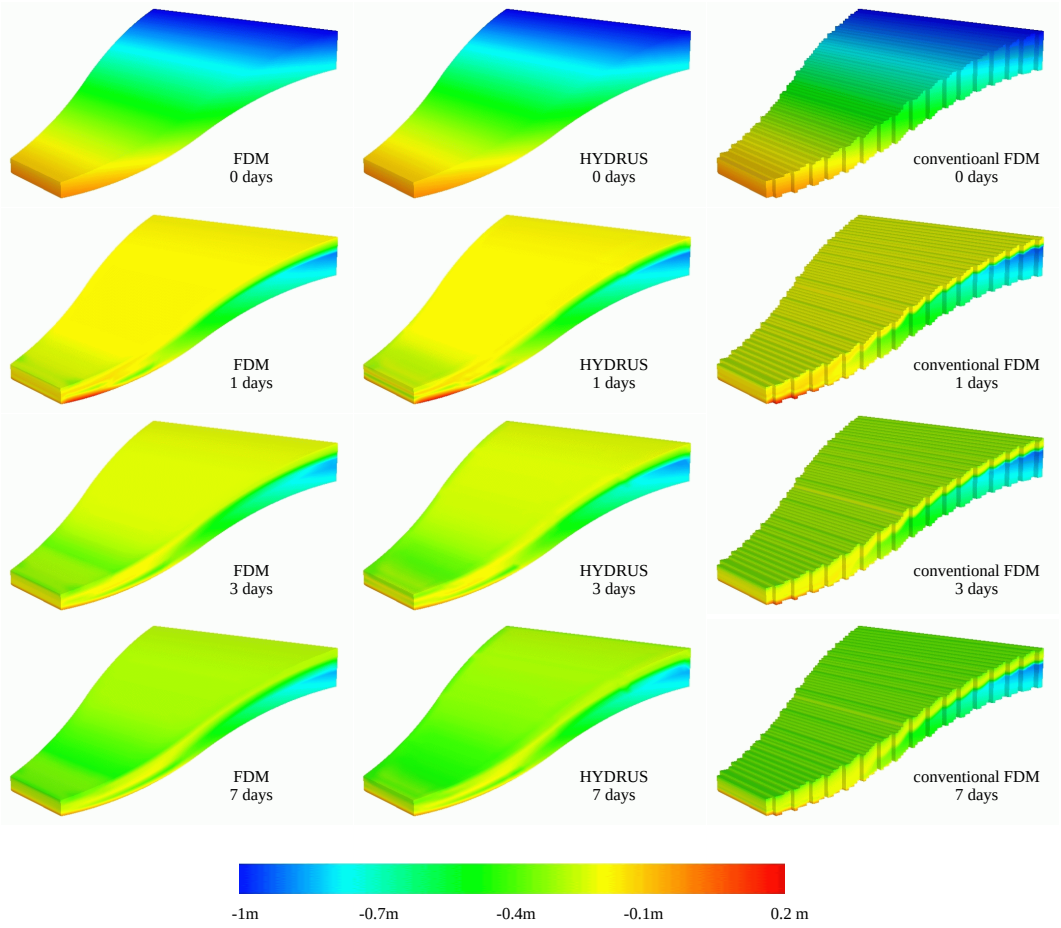

Figure 17. Pressure head results of Test 4; the left, middle, and right side show the results obtained by FDM, HYDRUS, and conventional FDM, respectively.

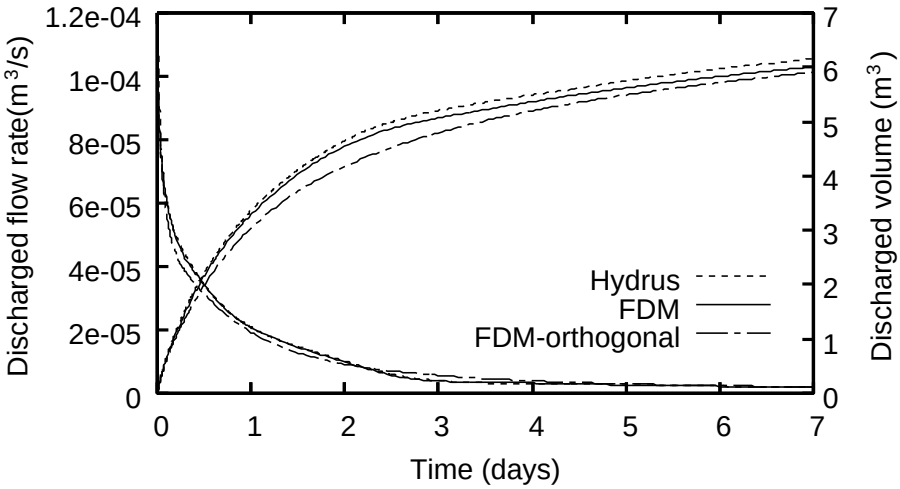

Figure 18. Discharge flow rate (thin line) and cumulative water volume (thick line) at the lower end of the slope in Test 4.

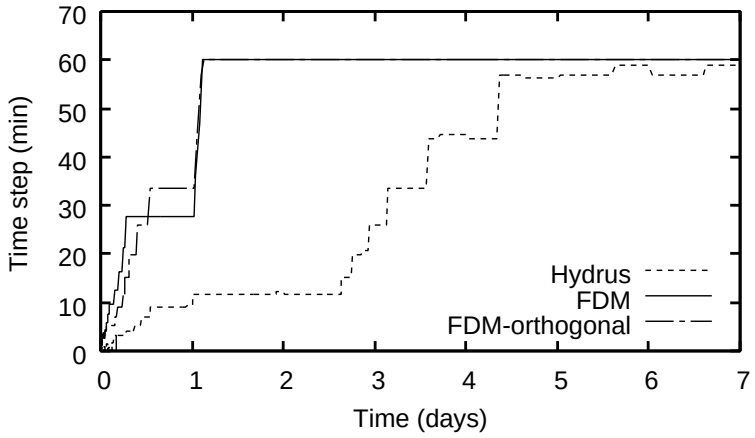

Figure 19. The time-step durations of FDM, HYDRUS, and conventional FDM for Test 4.

Table 1. Soil properties used for the Tests from *Carsel & Parrish* [1988]

Texture	θ_s m^3/m^3	θ_r m^3/m^3	K_s m/s	α m^{-1}	n
sand	0.43	0.045	8.250×10^{-5}	14.5	2.68
sandy loam	0.41	0.065	1.228×10^{-5}	7.5	1.89
loam	0.43	0.078	2.889×10^{-6}	3.6	1.56
silt	0.46	0.034	6.944×10^{-7}	1.6	1.37

Table 2. Test simulation 2, Relative error and mass balance error.

Grid	Number of cells	Relative error[a]			MBE[a] (%)		
		FDM	FI-FDM	HYDRUS	FDM	FI-FDM	HYDRUS
Grid1	20×20	6.09e-02	6.09e-02	5.66e-02	2.66e-02	2.66e-02	3.51
	20×40	2.29e-02	2.29e-02	1.78e-02	2.44e-02	2.44e-02	3.31
	20×80	7.17e-03	7.17e-03	5.46e-03	1.92e-02	1.92e-02	3.65
Grid2	20×20	4.89e-02	5.00e-02	6.90e-02	4.22e-03	5.66e-04	8.83e-01
	20×40	1.66e-02	1.76e-02	1.65e-02	3.42e-04	7.66e-04	4.20
	20×80	4.24e-03	4.86e-03	3.90e-02	1.72e-03	2.08e-03	11.56
Grid3	20×20	2.75e-02	3.02e-02	8.73e-02	4.57e-03	6.90e-04	3.32e-01
	20×40	1.50e-02	1.36e-02	2.83e-02	1.15e-04	4.69e-04	9.96e-01
	20×80	1.53e-02	1.22e-02	5.71e-03	2.01e-03	6.89e-04	3.94
Grid4	20×20	6.53e-02	5.88e-02	1.07e-01	1.28e-02	9.34e-04	6.01e-01
	20×40	6.86e-02	5.70e-02	3.00e-02	1.94e-04	3.27e-03	2.68
	20×80	7.00e-02	5.77e-02	5.40e-02	1.80e-03	9.56e-03	18.51

[a] Relative error and mass balance error are evaluated at the end of simulation.

Table 3. Test simulation 2, model performance.

Grid	Number of cells	CPU time (sec)			Iteration		
		FDM	FI-FDM	HYDRUS	FDM	FI-FDM	HYDRUS
Grid1	20×20	4	4	4	787	787	2074
	20×40	14	15	25	1279	1279	5959
	20×80	35	44	140	1843	1843	16555
Grid2	20×20	3	4	5	803	765	2421
	20×40	14	16	25	1470	1460	6278
	20×80	60	79	90	2959	2938	11254
Grid3	20×20	5	3	4	1007	602	1734
	20×40	16	14	21	1892	1247	5111
	20×80	75	77	109	3310	2766	13063
Grid4	20×20	5	3	5	1227	531	2479
	20×40	21	14	33	2502	1178	7679
	20×80	81	71	84	4242	2446	9976

Table 4. Test simulation 3, numerical accuracy and performance.

Model	Relative error[a]	MBE[a] (%)	CPU (sec)	Iteration
FDM	8.75e-02	6.99e-03	12	1692
HYDRUS	8.99e-02	0.32	36	7978

[a] Relative error and mass balace error are evaluated at the end of simulation.

Table 5. Test simulation 4, mass balance error and model performance.

Model	MBE[a] (%)	CPU (sec)	Iteration	Number of nodes
FDM	0.25	39	702	4896
HYDRUS	0.24	140	3347	4560
conventional FDM	0.48	577	678	34441

[a] MBE is evaluated at the end of simulation.